QED: A Fact Verification System for the FEVER Shared Task

Jackson Luken
Department of Computer Science
The Ohio State University
luken.25@osu.edu

Nanjiang Jiang, Marie-Catherine de Marneffe
Department of Linguistics
The Ohio State University
jiang.1879,demarneffe.1@osu.edu

Abstract

This paper describes our system submission to the 2018 Fact Extraction and VERification (FEVER) shared task. The system uses a heuristics-based approach for evidence extraction and a modified version of the inference model by Parikh et al. (2016) for classification. Our process is broken down into three modules: potentially relevant documents are gathered based on key phrases in the claim, then any possible evidence sentences inside those documents are extracted, and finally our classifier discards any evidence deemed irrelevant and uses the remaining to classify the claim's veracity. Our system beats the shared task baseline by 12% and is successful at finding correct evidence (evidence retrieval F1 of 62.5% on the development set).

"Supports" Claim: Ann Richards was professionally involved in politics.
Evidence set 1: Dorothy Ann Willis Richards (September 1, 1933 September 13, 2006) was an American politician and 45th Governor of Texas.
Evidence set 2: A Democrat, she first came to national attention as the Texas State Treasurer, when she delivered the keynote address at the 1988 Democratic National Convention.

"Refutes" Claim: Andrew Kevin Walker is only Chinese.
Evidence set: Andrew Kevin Walker (born August 14, 1964) is an American BAFTA-nominated screenwriter.

Figure 1: Claim/evidence examples from the FEVER data.

1 Introduction

The FEVER shared task (Thorne et al., 2018) sets out with the goal of creating a system which can take a factual claim and either verify or refute it based on a database of Wikipedia articles. The system is evaluated on the correct labeling of the claims as "Supports," "Refutes," or "Not Enough Info" (NEI) as well as on valid evidence to support the label (except in the case of "NEI"). Each claim can have multiple evidence sets, but only one set needs to be found so long as the correct label is applied. Figure 1 gives an example of a claim along with the evidence sets that support it, as well as a claim and the evidence that refutes it.

We split the task into three distinct modules, with each module building on the data of the previous one. The first module is a document finder finding key terms in the claim which correspond to the titles of the Wikipedia articles, and returning those articles. The second module takes each document found and finds all sentences which are close enough to the claim to be considered evidence. Finally, all sentences retrieved for a given claim are classified using an inference system as supporting or refuting the claim, or as "NEI". In the following sections, we detail each module, providing results on the FEVER development set which consists of 19,998 claims (6,666 in each class). Our system focuses on finding evidence sets composed of only one sentence. Of the 13,332 verifiable ("Supports" or "Refutes") claims in the development set, only 9% cannot be satisfied with an evidence set consisting of only one sentence. The code for our system is available at https://github.com/jluken/FEVER.

2 Document Finder

To verify or refute a claim, we start by finding Wikipedia articles that correspond to the claim. Key phrases within the claim are extracted and checked against Wikipedia article titles. If the key phrase matches an article title, the corresponding document is returned as potentially containing rel-

Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pages 156–160
Brussels, Belgium, November 1, 2018. ©2018 Association for Computational Linguistics

evant evidence to assess the claim's veracity.

2.1 Wiki Database Preprocessing

We created three maps of the Wikipedia article titles to deal with unpredictable capitalization and pages with a supplemental descriptor in the title via parenthesis (e.g., "Tool (band)" for the music group vs. the physical item.) The first map is simply a case-sensitive map of the document text mapped to its title. The second is titles mapped to lowercase. The third is a list of every title with a parenthesis description mapped to its root title without parenthesis. These are used as "backup" documents to be searched if no evidence is found in documents returned with the two other maps.

2.2 Key Phrase Identification

The key phrases aim at capturing the "topic" of the claim. We used capitalization, named entity, phrasal and part-of-speech tags, and dependency from the CoreNLP system (Manning et al., 2014) to identify key phrases. Subject, direct object, and their modifier/complement dependencies are marked as "topics". Noun phrases containing those topic words are considered key phrases. Consecutive capitalized terms, allowing for lowercase words not capitalized in titles such as prepositions and determiners, are also considered key phrases. For instance, the key phrases for the claims in Figure 1 are: *Ann Richards*, *politics*; *Andrew Kevin Walker*.

Once all possible key phrases in a claim are found, each key phrase is checked against the maps of Wikipedia titles: if there is a full match between a key phrase and a title, the corresponding article is returned. If the article found is a disambiguation page, each article listed on the page is returned. If the disambiguation page is empty, the results from the parenthesis map are returned.

2.3 Results and Analysis

On the development set, when only considering documents found using the case-(in)sensitive maps, we achieve 19.1% precision and 84.8% recall where at least one of the correct documents are found. However when the backup documents are also taken into consideration, recall raises to 94.2% while precision drops to 7.5%. The drop in precision is largely due to disambiguation pages, for which every document listed on the page gets returned. At this stage, we focus on recall, extracting as many relevant documents as possible (7.64 on average per claim), which will be filtered out in later stages.

Most of the 5.8% claims for which the system does not find any correct document involve noun phrases which CoreNLP fails to recognize (such as the song title *In the End*) as well as number mismatch between the claim and the Wikipedia article title (e.g., the system does not retrieve the page "calcaneal_spur" for the claim "Calcaneal spurs are unable to be detected in any situation.") Working on lemma could alleviate the latter issue.

3 Sentence Finder

Once all potential documents are collected by the Document Finder, each sentence within each document is compared against the claim to see if it is similar enough to be considered relevant evidence.

3.1 Claim Processing

The claim is processed to find information to check each document sentence against. We use the root of the claim and a list of all nouns and named entities in the claim. However, nouns and named entities included in the document's title are discarded from the list. This is done under the assumption that every sentence in a document pertains to the topic of that document (e.g., the second evidence in the "Supports" claim of Figure 1 from the document "Ann_Richards" refers to the subject without explicitly stating so.)

3.2 Extracting Evidence from Documents

A sentence is deemed potential evidence if it contains the root of the claim when the root is a verb other than forms of *be* and *have*.

We also retrieve sentences whose words sufficiently match the claim's list of nouns and named entities:

- If two or more are missing, the sentence is discarded.
- If all items in the noun and named entity list can be found in the document sentence, the sentence is added as evidence.
- If there is only one missing noun item, the sentence is added if there are at least two other matching items in the sentence, both the claim and document sentence are of the form "X is Y", or the document sentence contains a synonym of the noun, according to the MIT Java WordNet interface (Finlayson, 2014).

- If there is only one missing named entity, it can be swapped out with a named entity of the same label type. This allows to capture evidence for refuting a claim, such as mismatch in nationality (e.g., swapping out "American" for "Chinese" in the Andrew Walker example in Figure 1.) However, if a claim is centered around an action, determined by its root being a verb, an entity can be swapped only if the document sentence contains that same verb (or a synonym of the verb).

When the claim contains reference to either a birth or a death, the document sentence needs only to have a date encompassed within a set of parenthesis to be considered a valid piece of evidence.

3.3 Results and Analysis

Given a hypothetical perfect Document Finder (PDF), the Sentence Finder achieves a 51.9% precision and 50.3% recall on the development set. When using our existing Document Finder, precision drops to 24.0% and recall to 46.6%. However, we found that a number of sentences we retrieve are not part of the gold standard when they are in fact valid evidence for the claim. One such example is the sentence "As Somalia's capital city, many important national institutions are based in Mogadishu" to support the claim "There is a capital called Mogadishu." It is unclear how many examples of this there are.

If we evaluate the Sentence Finder on retrieving at least one accurate evidence for the verifiable claims, it achieves an accuracy of 66.2% with PDF, and 61.2% with ours. The Sentence Finder performs better on "Support" claims (70.73% with PDF) than on "Refutes" claims (61.58%).

On average using PDF, 1.14 sentences are returned for every claim for which evidence is found (51.9% of these being in the gold standard.) For 24.5% of the verifiable claims, the system fails to return any evidence (20.4% of "Supports" claims, 28.5% of "Refutes" claims.) Two of the most common causes of failure to retrieve evidence are mis-classification of named entity labels or part-of-speech tags by the CoreNLP pipeline, as well as an unseen correlation of key phrases between the claim and evidence based on context. For instance, our system fails to retrieve any of the evidence for the claim in Figure 1, missing the contextual connection between *politics* and *Governor* or *Democrat*.

4 Inference

Once evidence sentences are retrieved, we used one of the state-of-the-art inference systems to assess whether the sentences verify the claim or not. We chose the decomposable attention model of (Parikh et al., 2016) because it is one of the highest-scored systems on the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) that has a lightweight architecture with relatively few parameters.

4.1 Preprocessing

Most of the evidence sentences are often in the middle of a paragraph in the document, and the entity the document is about is referred to with a pronoun or a definite description. For instance, *The Southwest Golf Classic*, in its Wikipedia article, is referred to with the pronoun *it* or the noun phrase *the event*. We thus made the simplifying assumption that each pronoun is used to refer to the entity the page is about, and perform a deterministic coreference resolution by replacing the first pronoun in the sentence with the name of the page.

We ran a named entity recognizer trained on OntoNotes (Hovy et al., 2006) on claim and evidence sentences to extracted all the named entities and their types. The named entities concatenated with the sentence are fed into the word embedding layers whereas the named entity types are fed into the entity embedding layers, as described below.

4.2 Embedding

We used GloVe word embeddings (Pennington et al., 2014) with 300 dimensions pre-trained using CommonCrawl to get a vector representation of the evidence sentence. We also experimented training GloVe word embeddings using the provided Wikipedia data, but found that they did not perform as well as the pre-trained vectors. The word embeddings were normalized to 200 dimensions as described in (Parikh et al., 2016). For entity types, we trained an entity type embedding of 200 dimensions. The word embeddings and entity embeddings are concatenated together and used as the input to the network.

4.3 Training

All pairs of evidence/claim from the FEVER training data are fed into the network for training.

	Development set					Test set	
	Baseline	**Our system**				**Baseline**	**Our system**
	All	All	Supports	Refutes	NEI	All	All
FEVER score	31.3	43.9	54.9	24.7	52.0	27.5	43.4
Label Accuracy	51.4	44.7	68.6	31.3	52.0	48.8	50.1
Evidence Precision	N/A	77.5	77.0	78.1	–	N/A	N/A
Evidence Recall	N/A	52.3	56.3	47.8	–	N/A	N/A
Evidence F1	17.2	62.5	65.0	59.3	–	18.3	58.5

Table 1: Scores on the FEVER development and test sets. Baseline is the system from (Thorne et al., 2018). The results are prior to human evaluation of the evidence.

Since the "NEI" class does not have evidence associated with it, we used the evidence found by our Sentence Finder for training the "NEI" class. If our Sentence Finder did not return any evidence for a "NEI" claim, we randomly sampled five sentences from the sentences in the Wiki database and use them as evidence.[1]

The network is trained using the Adam optimizer with a learning rate 0.002 with a batch size of 140 and dropout ratio of 0.2. The network weights are repeatedly saved and we used the model performing best on the FEVER development set.

4.4 Assigning Class Labels

The network outputs a probability distribution for whether the evidence/claim pair has label "Supports", "Refutes", or "NEI". For a given claim, we examine the labels assigned for all evidence sentences returned for that claim. First, we discard the evidence labeled as "NEI". If there are no evidence left, we mark the claim as "NEI". Otherwise, we add together the remaining prediction distribution and use the highest scored label as label for the claim. We return the five highest-scored evidences, including those marked "NEI".

4.5 Results and Analysis

The resulting scores on the development and test sets are in Table 1 (prior to human evaluation of the evidence retrieved by the system.) The FEVER score is the percentage of claims such that a complete set of evidence is found and is classified with the correct label. Precision, Recall, and F1 are the metrics for evaluating evidence retrieval (evidence retrieval is not evaluated for the "NEI" class.)

Gold \ Labeled as	Supports	Refutes	NEI
Supports	68.59	2.87	28.55
Refutes	31.13	31.26	37.61
NEI	42.29	5.75	51.97

Table 2: Contingency matrix (percentage) in the development set.

Table 2 shows the percentage of claims being labeled as each class in the development set. We see that both "Refutes" and "NEI" are often mislabeled as "Supports", whereas the "Supports" are often mislabeled as "NEI".

Upon closer look at the classification errors, we see that some fine-grained lexical semantics and world knowledge are required to predict the correct label, which the model was not able to capture. For example, the claim "Gin is a drink" is supported by the sentence "Gin is a spirit which derives its predominant flavour from juniper berries (Juniperus communis)", but our system classified the pair as "Refutes".

The network also seemed to pick up on some lexical features present in the annotations. The claim "The Wallace mentions people that never existed" has gold label "NEI", but is labeled as "Refutes" with high probability using three different evidence sentences we retrieved, even though some of the sentences are not relevant at all. This is probably because the word "never" is highly indicative of the "Refutes" class, as we shall see in the next section.

5 Discussion

Our system beats the shared task baseline on evidence retrieval F1 (62.5% vs. 17.2%) and FEVER

[1] We could try the NEARESTP sampling method described in (Thorne et al., 2018), which achieves better performance with a decomposable attention model for inference than random sampling.

score (43.9% vs. 31.3%) for the development set. On the test set, prior to human evaluation of the evidence, our system ranked 7th out of 23 teams with a FEVER score of 43.4%. For evidence retrieval F1, we ranked 2nd with a score of 58.5%.

Gururangan et al. (2018) pointed out that natural language inference datasets often contain annotation artifacts. They found that many lexical/syntactic features are highly predictive of entailment classes in most natural language inference datasets. We performed the same analysis on the FEVER training set to see whether a similar pattern holds. We calculated the probability distribution of the length of the claims by tokens for each class. Contrary to Gururangan et al's results, all classes have similar mean and standard deviation sentence length. We also calculated the pointwise mutual information (PMI) between each word and class. We found that negation words such as *not*, *never*, *neither*, and *nor*, have higher PMI value for the "Refutes" class than for the other classes. This is similar to Gururangan et al.'s observation that negation words are strong indicators of contradiction in the SNLI dataset. The "Refutes" claims in the FEVER training data indeed show a high percentage of negation words[2] (13.9% vs. 0.1% for "Supports" and 1.3% for "NEI").

Another source of bias comes from the way evidence annotation in the gold standard has been created with humans manually verifying the claims in Wikipedia. As pointed out in Section 3.3, evidence automatically retrieved can be correct even though not present in the gold standard. The way a human fact-checks might be different from what a computer can achieve. It would be interesting to analyze the evidence correctly retrieved by the systems participating in the shared task but not present in the gold standard, to see whether some patterns emerge.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Mark Alan Finlayson. 2014. Java libraries for accessing the princeton wordNet: Comparison and evaluation. In *Proceedings of the 7th International Global WordNet Conference*, pages 78–85. H. Orav, C. Fellbaum, & P. Vossen (Eds.).

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of NAACL-HLT 2018*, pages 107–112.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 57–60.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: A large-scale dataset for Fact Extraction and VERification. In *Proceedings of NAACL-HLT 2018*, pages 809–819.

[2] We used the `neg` dependency tag as the criterion for a negation word.